Audit

This module introduces the concept of audit, why it is necessary, and what events are typically audited. It describes the TCSEC requirements foraudit as well as various audit and audit analysis mechanisms that can be used to satisfy the TCSEC requirements.

Module Learning Objectives

The material presented in this module can be read independently of theother modules. Upon completion of this module, the student should:

1. Understand what an audit mechanism in a trusted system is.
2. Understand what events are typically audited.
3. Understand the TCSEC requirements for audit.
4. Be familiar with various audit mechanisms that can be used to satisfy the TCSEC requirements.
5. Be familiar with the capabilities of audit analysis and data reduction tools that are used to examine audit trails.

Overview

Auditing is used to record “who did what” in the audited system. For TCSEC classes C2 and above, it requires that all users' actions be open to scrutiny by means of active auditing. The TCB is required to “be able to create, maintain, and protect from modification or unauthorized access or destruction anaudit trail of accesses to the objects it protects.” The purpose of recording, examining, and reviewing all security-relevant events (as defined in the TCSECglossary) is to detect and deter penetration of a computer system, reveal usagethat identifies misuse by authorized users, and assist in the assessment of damages caused by a penetration or misuse of the computer system.

The TCSEC requirements for auditing are driven by the TCSEC's Accountability Control Objective, which requires individual accountability whenever either a mandatory or discretionary security policy is invoked.The TCSEC relates the Accountability Control Objective to auditing to createthe following control objective for auditing:

> “A trusted computer system must provide authorized personnel withthe ability to audit any action that can potentially cause access to, generation of, or effect the release of classified or sensitiveinformation. The audit data will be selectively acquired based on the auditing needs of a particular installation and/or application. However, there must be sufficient granularity in the audit data to support tracing theauditable events to a specific individual who has taken the actions or on whose behalf the actions were taken.”

The TCSEC audit requirements start at C2 by stating that the TCB must securely maintain an audit trail, and must describe the events that areto be audited and the parameters that are to be collected. At B1, the audit mechanism must also record the sensitivity label of objects accessed andany override of output markings. At B2, another requirement is added for detecting

covert channel exploitation. At B3, the TCB must have a monitoring mechanism that examines auditing information and notifies the security administrator or takes autonomous action if violation of the securitypolicy appears to be occurring. There are no additional requirements at A1.

The ability to select what is to be audited based on the needs of theinstallation may be met by pre-selection (collecting only the audit data that was chosen), or post-selection (collecting audit data on all actions but only viewinga subset). The trade-offs between the two methods are: (1) space requirementsof a complete audit trail versus the hazard of being unable to viewinformation that may have been mistakenly considered irrelevant in the pre-selection process, and (2) acceptance of system performance versus system degradation.Care should be directed at specifying the action to be taken in the event thatthe audit trail exhausts its primary storage medium.

The TCSEC and [AUDIT88] specify for each class which events are required to be audited and which particular data items are required to be collected. The TCB must protect the audit trail software, the audit trail itself, and the audit enabling/disabling mechanism. There must not be a way for an intruder to avoid leaving a transcript of his activity by either turning the auditing off, bypassing the audit process, or modifying the audit trail after the fact. Also, unauthorized users should not be permitted read access to the audit trail.

Standard operating system auditing may be deficient in meeting TCSEC requirements for auditing. Generally, auditing implemented on commercial operating systems, such as UNIX, are intended primarily for accounting purposes and will provide insufficient focus and detail for securitypurposes . It may be impossible to audit object accesses or failed system resourceaccess attempts (such as failed login attempts), and the auditing mechanism itself may not be sufficiently protected to prevent compromise.

As indicated in the required readings for this module, the audit requirements have been the subject of many Interpretations. The Interpretations do not change the TCSEC audit requirements; they simply explain therequirements . In some cases, the explanations are only for a specific context. The followinglist briefly summarizes the Interpretations of the audit requirements:

1. The ability to audit all security-relevant actions and events must be present in order to pass the audit requirements at any TCSEC class.
2. The ability to audit “object not found” is not required at C2 or B1.It is required at B2 and above if it results in covert channels.
3. The audit mechanism must be sufficiently flexible for the system administrator to obtain information regarding system activity based upon user's identity or sensitivity label. Audit reduction tools must be maintained under the same configuration control system as the remainder of the system.
4. The audit mechanism must provide bounds checking on counts and rates of events which may violate the security policy.
5. The audit trail must completely and accurately reflect most actions taken by users that have been specified to be recorded in the audit trail.

6. Level changes on single-level communication channels and I/O devices must be auditable. At B1, label changes on multilevel communication channels and I/O devices are not required to be auditable. Above B1, changes in device ranges must be auditable.
7. If the TCB supports the selection of events to be audited, it shallprovide a method for immediate enforcement of a change in audit settings (e.g., to audit a specified user, to audit objects with a particular sensitivity label); however, the immediate method (e.g., shutting the system down) need not be the usual method for enforcement. The TFM shall describe both the usual enforcement of audit settings and, if the immediate enforcement method is different from the usual one, how an administrator can cause immediate enforcement.
8. When the TCB becomes unable to collect audit data, it shall give a clear indication of this condition and take a pre-specified action. Thesystem implementation may allow an administrator to choose from a range of actions. One of the actions that may be chosen by the administrator (or, if no choice is possible, the only action) shall be that the system cease performing auditable events when the TCB is unable to collect audit data. Choosing an audit overflow action shall be considered asecurity-relevant administrative event for the purposes of auditing. The TFM shall fully describe the administrator's options.
9. While the audit mechanism is required to be capable of producing a record of each login attempt, the audit trail is not required to record the character string supplied as the user identity on failed login attempts.
10. Audit of imminent violations is required only for actions that may lead to an actual violation (i.e., multiple failed login attempts, excessive use of identified auditable covert channels). The threshold for notification of the security administrator may be either set by an administrator or fixed by the vendor. There shall be an immediate notification mechanism included with the product (e.g., a message to an identified terminal, a red light as part of the hardware), and the TFM shall describe how the security administrator can monitor this mechanism.
11. At B3 and A1, the action taken to terminate an event leading to an imminent violation must at least eliminate the capability to repeat the event. A convincing argument must be made as to why the chosen action is “least disruptive.”

The development of an auditing subsystem for a compartmented mode workstation implemented on a UNIX operating system is described in [Picciotto87]. This work demonstrates that a comprehensive audit capability may be implemented on an existing operating system without affecting the function of commercial-off-the-shelf applications. The following auditing subsystem requirements are identified in [Picciotto87]

1. Accept audit data from processes trusted to generate their own audit records.
2. Collect sufficient audit data to deduce all command informationentered by system users.

3. Collect audit data related to security-relevant events, including object access.

To develop a security audit capability by modifying an existing operating system, there are four necessary coding efforts: (1) introducing audit probes at the kernel's system call level, (2) introducing audit probes at the user/operating system interface level, (3) supporting auditing generated fromwithin application programs, and (4) providing a user interface to the audit subsystem. Auditing at the user/operating system interface level andfrom within the operating system itself is generally sufficient to meetsecurity-related requirements. However, the addition of application level auditing works to reduce the volume of audit data collected, making audit trailseasier to comprehend by auditing at a higher level of abstraction. For example, a kernel level auditing of the editing of a file may include many uninteresting internal word processor actions which would cloud the user's actual intentions. In order to reduce the auditing of these actions trusted applicationsshould have the right to generate their own audit records.

Care must be taken that a system modified to support security auditingdoes not become so cumbersome as to become useless. Techniques such as data compression or selective auditing may be used to limit the volume of data generated by the system. It should be expected that performance of disk-intensive processes will be greatly degraded because most auditableactions are concerned with file system manipulation.

The overall goal of a security audit system is to provide authorizedsystem personnel with a means to regularly review a documented history ofselected activity descriptions on the system. If a system violation should occur, this documented history will log and permit reconstruction of the eventsleading up to and including the violation. The documented history also permits surveillance of users' activity. The surveillance of users' activity in this manner may provide notice of (unsuccessful) attempts to violate systemsecurity so that the anomalous activity may be acted upon in advance of a successful violation.

Audit data often provides minimal real-time or near real-time benefitbecause the quantity and complexity of the data prohibits meaningful manual interpretation. The audit trails are instead only used after the factwhen an intrusion is suspected for other reasons (e.g., a user reports a file has been maliciously modified, or a system operator happens to notice unusualmodem activity). Only then are the audit trails carefully examined to assessdamage and accumulate evidence.

Sophisticated intrusions and insider abuse may not even be discernible by single incriminating events and would escape even concentrated manual analysis. Only with the aid of automated analysis tools can a system security officer effectively detect a pattern of evidence establishingundesirable activity. Automated audit analysis tools may be developed with a wide range of sophistication and degree of autonomy. The following three tool groupings are presented in order of increasing sophistication.

Audit Reduction Tools

The most straightforward and easiest audit analysis tool to develop is a simple set of analysis utilities that allow an Information System SecurityOfficer (ISSO) to better digest audit data by acting as a data reductiontool. Innocuous audit data would be eliminated and the filtered and formatted datadispla yed. The importance of audit data would be determined based on a crudedefinition , such as any commands that do not modify secured data are to be ignored. Analysis of the remaining data would be largely performed as before bymanual review. An approach to audit reduction/selection is described in [Sibert88].

Manually Trained Audit Analysis Tools

A more useful tool could be developed to perform statistical analyseswhic h would isolate obvious deviations from earlier learned thresholds fornormal activity (e.g., a user suddenly issuing a barrage of commands at the rate of some number per second who previously had been described to the tool as having an average command rate of one per minute would be flagged). The ISSO now needs only look at a much reduced statistically deviant subset of the audit trail. The results of a feasibility demonstration of such a tool that ranked UNIX sessions by degree of suspiciousness is described in [Halme86].This early research demonstrated that even very limited audit trails can be successful in flagging deviant activity.

Autonomous/Adaptive Audit Analysis Tools

The tools discussed in the previous two groupings are strictly anomaly detectors that might be run in batch at the end of the day. The values representing the specified norms for each user are considered fairly static and are reevaluated only on some limited amount of “training” datawhen directed by a human. A product of greater sophistication would be a nearreal-time tool that could run autonomously and would take prescribed action as serious threats to the system occur. Such a tool, known as an intrusion detection tool, would support self-maintaining adaptive profiling of each user's previous system usage. Any event that deviated from a particular user's normal work pattern would be flagged. The tool would intelligently derive thesecurity threatening severity of a system action by combining the severity ofindividual feature deviations. The severity of system actions would in turn be combined in a counter attached to individual users. The severities per user are ranked for ease of review by the ISSO.

Intrusion detection/countermeasure research is described in a number of conference papers: [Halme88], [Lunt88a], [Denning87], [Sebring88],[Clyde87], [Smaha88], [Vacarro89], and [Winkler89].

Relevant Trusted Product Questionnaire Questions

2.8 AUDIT

C2:

1. Provide a brief description (preferably in block diagram form) of audit data flow in terms of how the data are created, transmitted, stored, and viewed for analysis.
2. How are the audit logs protected?
3. (a) How can the audit log be read? (b) Who can invoke these mechanisms? (c) What privileges are required to invoke these mechanisms?
4. (a) What tools are available to output raw or processed (i.e., analyzed and reduced) audit information? (b) Who can invoke these tools? (c) What do the tools do in terms of audit data reduction? (d) What are the formats of the reports/outputs generated by these tools?
5. (a) How can the audit log be written or appended? (b) Whocan invoke these mechanisms? (c) What privileges are required to invoke these mechanisms?
6. (a) How can the audit log be deleted? (b) Who can invokethese mechanisms? (c) What privileges are required to invoke these mechanisms?
7. What are the internal formats of audit records?
8. Provide a list of auditable events (examples include attempted logins, logouts, creation of subjects, deletion of subjects, assignment of privileges to subjects, change of subject privileges, use of privileges by subjects, creation of objects, deletion of objects, initial access to objects (introduction of the object into a user's address space), assumption of the role of security administrator).
9. (a) Which actions by the trusted users are auditable? (b)Which are not? (Examples of trusted users are system operator, account administrator, system security officer/administrator, auditor, system programmer, etc. Trusted users almost always have at least one privilege.)
10. (a) What data are recorded for each audit event? (b) Which ofthe following data (if any) are not recorded for each event: date,time, user, object, object DAC information (e.g., ACL), type of event, invoked or not invoked, why not invoked, success or failure in execution, terminal identification?
11. (a) Can the password ever become part of the audit record?(b) If yes, under what circumstances can this occur?
12. (a) What mechanisms are available to designate and change the activities being audited? (b) Who can invoke these mechanisms? (c) What privileges are needed to invoke these mechanisms?
13. (a) What mechanisms are available for selective auditing (i.e., selection of events, subjects, objects, etc., to be audited)?(b) What parameters (e.g., individual or group of subjects, individual

objects, subjects within a sensitivity range, objects within a sensitivity range, event type) or combination of parameters can be specified for the selective auditing? (c) Who can invoke these mechanisms? (d) What privileges are needed to invoke these mechanisms?

14. When do changes to the audit parameters take effect (e.g., immediately for all processes, for new processes)?

15. (a) Are the audit reduction tools part of the TCB? (b) Ifnot, what trusted mechanism is used to view/output the audit log?

16. (a) Does the system produce multiple audit logs? (b) If yes,what tools, techniques and methodologies are available to correlate these logs?

17. (a) Who (e.g., operator, system administrator or other trusted user) is notified when the audit log gets full? (b) Whatoptions are available to handle the situation?

18. What other action does the TCB take when the audit log becomes full (e.g., halt the system, do not perform auditable events, overwrite oldest audit log data).

19. (a) In the worst case, how much audit data can be lost (e.g.,when audit log overflows, system goes down with audit data in memory buffers)? (b) Describe the worst case scenario. (c) Whencan it occur?

B1:

20. Which of the following events are auditable: change in the device designation of single level or multilevel, change in device level, change in device minimum or maximum level, override of banner page or page top and bottom markings?'

21. Are the (a) subject and (b) object sensitivity levelrecorded as part of the audit event?

B2:

22. Are events that exploit covert storage channels auditable?

B3:

23. How does the TCB (a) designate and (b) change the occurrenceor accumulation of events that require real-time notification? (c) Who can invoke these mechanisms? (d) What privileges are needed to invoke these mechanisms? (e) Who (e.g., system administrator, president of the company) gets the real-time notification? (f) What actions/options are available to the individual being notified? What does the TCB do about (g) the event and (h) the process that caused this alert?

Required Readings

TCSEC85 National Computer Security Center, *Department of Defense Trusted Computer Security Evaluation Criteria*, DoD 5200.28-STD, December 1985.

Sections 2.2.2.2, 3.1.2.2, 3.2.2.2, 3.3.2.2, and 4.1.2.2 contain the audit requirements, which are summarized on page 96.

INTERP94 National Computer Security Center, *The Interpreted TCSEC Requirements*, (quarterly).

The following Interpretations are relevant to audit:

I-0004	Enforcement of audit settings consistent with protection goals
I-0005	Action for audit log overflow
I-0006	Audit of user-id for invalid login
I-0040	Requirements for overwrite label capability
I-0043	Auditing use of unnamed pipe
I-0084	Audit least disruptive action
I-0073	OK to audit decision regardless of whether action completed
I-0172	Audit of imminent security violations
I-0286	Auditing unadvertised TCB interfaces
C1-CI-04-84	Audit
C1-CI-07-84	Audit
C1-CI-02-85	Audit
C1-CI-02-86	Server
C1-CI-02-87	Audit
C1-CI-01-88	Exportation of Labels
C1-CI-01-89	Audit
C1-CI-02-89	Audit
C1-CI-03-89	DAC Public Objects

AUDIT88 National Computer Security Center, *A Guide to Understanding Audit in Trusted Systems*, NCSC-TG-001, Version 2, 1 June 1988.

This document provides guidance on TCSEC audit requirements and discusses issues involved in implementing and evaluating an audit mechanism. Manufacturers are given advice on how to design and incorporate an effective audit mechanism into their system, and implementors are given advice on how to make effective use of the audit capabilities provided by trusted systems.

Halme88 Halme, L.R. and Kahn, B.L., "Building a Security Monitor with Adaptive User Work Profiles," *Proceedings of the 11th National Computer Security Conference*, pp. 274-283, October 1988.

This paper presents issues relevant to construction of audit-analyzing intrusion countermeasure equipment that runs autonomously and takes prescribed action as anomalous system

usage was detected. Such a tool would support self-maintaining adaptive work profiles of each user's previous system usage.

Lunt88b Lunt, T., "Automated Audit Trail Analysis and Intrusion Detection: A Survey," *Proceedings of the 11th National Computer Security Conference*, pp. 65-73, October 1988.

This paper overviews early audit analysis research. It surveys a number of the subsequent automated audit analysis and intrusion detection efforts.

Supplemental Readings

Sebring88 Sebring, M., Shellhouse, E., and Whitehurst, R.A., "Expert Systems in Intrusion Detection: A Case Study," *Proceedings of the 11th National Computer Security Conference*, pp. 74-81, October 1988.

This paper discusses the Multics Intrusion Detection and Alerting System (MIDAS) that was implemented on NCSC's Dockmaster machine.

Sibert88 Sibert, W.O., "Auditing in a Distributed System: SunOS MLS Audit Trails," *Proceedings of the 11th National Computer Security Conference*, pp. 82-90, October 1988.

This paper describes the SunOS MLS auditing mechanism and how it addresses the problems of performing useful audit functions in large distributed systems. An audit reduction tool is described.

Other Readings

Clyde87 Clyde, A.R., "Insider Threat Identification Systems," *Proceedings of the 10th National Computer Security Conference*, pp. 343-356, September 1987.

Denning87 Denning, D., "An Intrusion Detection Model," *IEEE Transactions on Software Engineering*, Vol. SE-13, No. 2, pp. 222-232, February 1987.

Dewy78 Dewy, R.H., "System Auditability and Control in an EFTS Environment," *AFIPS Conference Proceedings*, Vol. 47, 1978.

Halme86 Halme, L.R. and Van Horne, J., "Automated Analysis of Computer System Audit Trails for Security Purposes," *Proceedings of the 9th National Computer Security Conference*, pp. 71-74, September 1986.

Ilgun93 Ilgun, K., "A Real-Time Intrusion Detection System for UNIX," *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pp. 16-28, May 1993.

Lunt88a Lunt, T. and Jaganathan, R., "A Prototype Real-Time Intrusion-Detection Expert System," *Proceedings of the 1988 IEEE Symposium on Security and Privacy*, pp. 59-66, April 1988.

Lunt92 Lunt, T.F., Jaganathan, R., Lee, R., Listgarten, S., Edwards, D.L., Neumann, P.G., Javits, H.H., and Valdes, A., *A Real-time Intrusion Detection Expert System (IDES)*, SRI Technical Report, February 1992.

Perry78 Perry, W.E. and Warner, H.C., "Systems Auditability: Friend or Foe?," *The Journal of Accountancy*, February 1978.

Picciotto87 Picciotto, J., "The Design of an Effective Auditing Subsystem," *Proceedings of the 1987 IEEE Symposium on Security and Privacy*, pp. 13-22, May 1987.

Rahden79 Rahden, H.R., "Computer Security Auditing," *WESCON 1979 Conference Record*, 14/3, 1979.

Schaefer89 Schaefer, M., Hubbard, B., Sterne, D., Haley, T.K., McAuliffe, J.N., and Wolcott, D., "Auditing: A Relevant Contribution to Trusted Database Management Systems," *Proceedings of the 5th Annual Computer Security Applications Conference*, p. 232, December 1989.

Smaha88 Smaha, S.E., "Haystack: An Intrusion Detection System," *Proceedings of the 4th Aerospace Computer Security Applications Conference*, pp. 37-44, December 1988.

Vacarro89 Vacarro, H. and Liepins, G.E., "Detection of Anomalous Computer Session Activity," *Proceedings of the 1989 IEEE Symposium on Security and Privacy*, pp. 280-289, May 1989.

Winkler89 Winkler, J.R. and Page, W.J., "Intrusion and Anomaly Detection In Trusted Systems," *Proceedings of the 5th Aerospace Computer Security Applications Conference*, pp. 39-45, December 1989.